# APPLICATION FOR UNITED STATES LETTERS PATENT

INVENTOR(S): Jürgen ANGELE

TITLE: COMPUTER SYSTEM

ATTORNEYS:

VENABLE LLP
1201 New York Avenue, NW
Suite 1000
Washington, DC 20005-3917
Telephone: 202-962-4800
Telefax: 202-962-8300

ATTORNEY REFERENCE: 37934-191592

## CROSS-REFERENCE TO RELATED APPLICATION

**0001** This application is a continuation-in-part of PCT International Application No. PCT/EP02/05347, filed May 15, 2002, incorporated herein by reference, which claims the priority of German Patent Application No. 101 23 959.9, filed on May 17, 2001.

## FIELD OF THE INVENTION

**0002** The present invention addresses computer systems and methods for the efficient storage and recovery of information.

## DESCRIPTION OF RELATED ART

**0003** Computer systems of the type with which the present invention is concerted are often formed with computer units connected into a network and provided with means for storing data, in particular integrated database systems. In particular, the computer units can also be connected to the Internet, so that the database systems can be queried via the Internet.

**0004** Database systems of this type generally contain large data records that can be queried with specified query

commands. One critical problem, particularly with databases storing large amounts of data, is the definition of suitable query commands to obtain the desired search results.

**0005** Searches of this type are particularly difficult if information on general subjects is needed, but very little searchable data on these subjects is available.

**0006** As an example, suppose that a computer system comprises database systems in which different types of technical, medical and business publications are stored. A user of this database system knows the name of an author of a publication, but only knows that the publication is a technical publication. The only search term available to the user is the name of the author of this publication. This name functions as input variable for the search, which is input via a query unit into the computer system. The user must search through all query results for the researched name to reach the desired publication, if necessary by using additionally available information on the author, because no further information is available. An additional manual evaluation of this type is extremely involved and is also the cause of many error sources, so

that the search result is subject to considerable inaccuracies.

## SUMMARY OF THE INVENTION

**0007** It is an object of the present invention to create a computer system of the aforementioned type and an associated method to ensure the most comprehensive, simple and flexible access to information stored in the computer system.

**0008** According to one embodiment, the invention comprises a computer system that includes means for storing data; means for allocating the data to classes of at least one class structure forming an object model; an inference unit for generating output variables by evaluating rules; means for the input of a query command for instructing said inference unit to generating output variables by evaluating rules, the rules forming a declarative system and linking components of class structures; and an editor for generating at least one of rules, class structures, and components thereof.

**0009** According to a further embodiment, the invention comprises a method of storing and retrieving data in a computer system. The method may comprise the steps of

forming at least one object model, the object model including at least one class structure; allocating data according to one or more classes of said at least one class structure; providing a set of rules, the rules forming a declarative system and linking components of class structures; providing a query command; and in response to the query command, processing a series of said rules to obtain one or more output variables.

**0010** The invention may further be embodied in the form of a computer-readable medium containing software code embodying the above method. Examples of such computer-readable media include RAM, ROM, PROM, bubble memory, magnetic media, the various types of disks, paper tape, punch cards, etc.

**0011** The invention may also be embodied in the form of a computer system executing the above method or comprising a processor and the above computer-readable medium. It may further be embodied in the form of a data signal carrying software embodying the method.

**0012** The basic idea behind the invention, therefore, is that the data stored in the computer system are structured within at least one object model, preferably within several object models. Object models of this type, which form ontologies, represent class structures having classes that

are structured hierarchically or in acyclical graphs, wherein several attributes that are passed on within a class structure are preferably allocated to the classes. According to the invention, information stored in the computer system can not, or not solely, be accessed by querying data stored therein. Instead, the computer system according to the invention comprises a predetermined number of rules that are allocated to at least one inference unit.

**0013** Attributes of at least one class structure and/or classes of at least one class structure and, if necessary, also stored data can be linked with these rules. The rules here represent the logical linking rules, which relate the individual aforementioned elements in a pre-defined manner to each other. An evaluation is made in the inference unit where concrete values for the attributes, classes and/or data are allocated to these rules, thus generating specific output variables.

**0014** With the editor according to the invention, the computer system user him- or herself can generate the rules and/or class structures. For this, the editor is provided with corresponding programming surfaces or graphic surfaces by means of which the respective components of the rules and/or class structures can be input. Thus, the rule

structures as well as the class structures can be input and changed easily and flexibly with the editor.

**0015** In particular, graphic surfaces can be provided for generating the rules, and which make it possible to define the rules completely. Alternatively, rules can be programmed freely in the editor, meaning the user specifies all the rules through the input of programming commands. According to another advantageous embodiment of the invention, axioms that are specified in the editor can be selected. If the user selects an axiom, a rule that is clearly allocated to this axiom and is preferably stored in the editor is activated, thus making it particularly easy for the user to generate the desired rules.

**0016** One essential advantage of the computer system according to the invention is that the query and evaluation of information is not limited to the level of the data stored in the computer system. Rather, the queries are expanded to include the structural elements of the object models for structuring the data. Thus, even complex facts and connections can be extracted from the information stored in the computer system even with rudimentary and simple queries. and/or input values.

**0017** A simple example for the computer system according to the invention is a computer unit with an integrated database system. By querying specific classes or attributes, data subsets are obtained as output variables without having to query the data itself directly. Query schemes of this type are particularly advantageous because they make it possible to classify data according to easily searched specific criteria and characteristics by using the classes and attributes of class structures. These sub-classes can then be divided further into additional subordinate classes, to which various attributes are allocated.

**0018** The data allocated to these elements can then be determined by querying specific classes and attributes with the computer system according to the invention, without entering a concrete query for concrete data. A particularly powerful and flexible query system that considerably expands the search options as compared to traditional database systems is thus created with the option of querying the class and/or attribute level above the data level.

**0019** Furthermore, relationships can be established between data, attributes and/or classes not stored in this form in

the database system by evaluating the rules. Thus, new knowledge can be derived from known, stored variables by using the computer system according to the invention.

**0020** Another advantage of the computer system according to the invention is that a user can input several query terms as input variables into the computer system without having to make a distinction whether these query terms are data, classes or attributes to be searched. In the form of input variables that are entered into the inference unit, these query terms are linked to the respective rules allocated to a query command. The query terms are then allocated, with the aid of the rules, to the data, classes and/or attributes of an object model. Data subsets are obtained as output variables, which have a predetermined relationship to each other, corresponding to the setup of the rules. In the simplest case, the query terms are linked to form a single output variable.

## BRIEF DESCRIPTION OF THE DRAWINGS

**0021** The invention is explained in the following with the aid of the attached drawings, in which:

**0022** Figure 1 shows a configuration of an exemplary embodiment of the computer system according to the invention;

**0023** Figure 2 shows object models for structuring the data stored on a computer system according to Figure 1; and

**0024** Figure 3 shows an exemplary embodiment of a graphic surface for the editor for the computer system according to Figure 1.

## DETAILED DESCRIPTION OF THE INVENTION

**0025** Figure 1 contains a schematic diagram of the basic design of a first exemplary embodiment of the inventive computer system 1. The computer system 1 comprises several computer units 2 that are connected via computer lines 3 to form a network. One of the computer units 2 functions as a central computer on which data are stored. A database system 4 forms the means for storing the data. An inference unit 5 is provided for carrying out and evaluating queries in the database system 4. In addition, the computer unit 2 comprises a terminal 6 as input/output unit, which can be used to operate an editor.

**0026** Several users can access the database system 4 via additional computer units 2 that are connected to the

network. These computer units 2 may be, for example, personal computers. The computer units 2 are provided with suitable input/output units for this, which preferably take the form of terminals 6.

**0027** The Internet in particular may function as the network, in which case the computer units 2 are provided with corresponding Internet hookups.

**0028** Object models, so-called ontologies, are used for structuring the data stored in the database system 4. An object model has a class structure, wherein the structure may be a hierarchical structure. With hierarchical structures, the classes of a predetermined level are respectively allocated to precisely one class of a superior level, meaning only single inheritances are permitted. In general, the class structure can also be embodied as an acyclic graph for which multiple inheritances are permitted.

**0029** Figure 2 shows examples of two such hierarchical class structures that respectively form one object model. The first object model contains a "publications" class to which the subordinate classes "lectures" and "documents" are allocated. The second object model contains a "persons" class to which the subordinate classes "non-employees" and

"employees" are allocated, in which the additional subordinate classes "technical employees" and "clerical employees" are allocated to the sub-class "employees."

**0030** Specific attributes are allocated to the classes of respectively one hierarchical class structure. In the process, an attribute allocated to a class, for example, the "persons" class, is passed onto the subordinate classes for this class, wherein, for example, an attribute of this type can be a name. For the present example, this attribute is passed on within the class structure to the subordinate classes "non-employees" and "employees," as well as to the subordinate classes for this latter class, meaning "clerical employees" and "technical employees." In this way, a particularly efficient structuring of the data in the database system 4 is consequently created.

**0031** Rules are allocated to the inference unit 5 for processing the queries in the database system 4. These rules are stored in the inference unit 5 itself or in a memory unit that is allocated to the inference unit 5 (and not explicitly shown in Figure 1).

**0032** The object models as well as the language for formulating these rules can differ. The object models are preferably of the type DAML+OIL (where DAML stands for

"DARPA Agent Markup Language, a language/tools for facilitating the concept of the semantic web), wherein DAML-L is used as the rule language.

**0033** To process queries in the database system 4, defined query commands are entered into an input/output unit 6. Depending on the format for the query command, a series of rules is processed in the inference unit 5. Since the rules in general are a declarative system, the sequence for the definition of the rules is not important.

**0034** The rules involve relationships in the form of logic links between classes and/or attributes and/or data of the database system 4. The rules allocated to a query command for generating defined output variables are evaluated in the inference unit 5. It is useful if the output variables are subsequently output via the input/output unit.

**0035** By linking attributes and classes via a predetermined number of rules, it is easy to query data subsets in the database system 4, without having to refer to specific data in the query commands.

**0036** As compared to traditional database systems 4, where the query commands are limited to the data level, the option of processing a query on the class and attribute

level allows for a considerable expansion and higher flexibility of the processing options.

**0037** For example, a query command of this type can have the following format:

**0038** "Output of the names for all data stored in the hierarchy of the class structure for the object model "persons" below the "employees" level.

**0039** The names of all technical and clerical employees stored in the database system 4 are then displayed for the user as output variables.

**0040** According to another advantageous embodiment, relationships between different attributes, classes and/or data can be created with the rules allocated to the individual query commands. In particular, attributes, classes and/or data from the various class structures can also be linked with these rules.

**0041** A particular advantage is that the user only needs to input the terms for processing the search, preferably in sequence, when entering the query command. The user is not required to define whether these terms relate to classes, attributes or data. In addition, the user is not required to intervene in the structure of the rules that are allocated to a specific query command. The inference unit

5 automatically allocates the terms to the rules and processes the rules.

**0042** One example of a query of this type can be structured as follows. A user would like to inquire about the level of knowledge of a person, known to the user, with the name "Mustermann."

**0043** The user thus enters the two search variables "Mustermann" and "knowledge" into the input/output unit.

**0044** The rules allocated to this query command are evaluated in the inference unit 5, wherein a rule of this type can be worded as follows:

**0045** "If a person writes a document and the document deals with a subject, then this person has knowledge of the subject."

**0046** The classes "persons" and "document" from two different class structures are linked with this rule. Reference is made in the process to the subject of documents, wherein the subjects of documents, for example, are allocated as data to the "document" class.

**0047** Whether or not a person has "knowledge" of this subject is obtained as an output variable for this rule.

**0048** The example shows that the query not only obtains information stored in the database system 4 as a result of

such links. Rather, rules of this type establish relationships between elements in the database systems 4, such that new characteristic variables can be derived if necessary. It means that with the rules from the data stored in the database system 4, new knowledge can be derived, which is not stored in this form in the database system 4.

**0049** In the inference unit 5, the above-mentioned rule is evaluated in dependence on the input variables "knowledge" and "Mustermann" with the aid of an allocation diagram stored therein, which reads as follows for the present case:

Mustermann is a person.

Mustermann is the author of a dissertation.

The subject matter of this dissertation is biotechnology.

The dissertation is a document.

**0050** Using the aforementioned rule for evaluating these allocations results in showing that "Mustermann" has knowledge of biotechnology. The result is preferably output via the input/output unit.

**0051** One essential difference between this system and known database systems 4 is that the search result "Mustermann

has knowledge of biotechnology" was not obtained either through a query of the database with the term "knowledge" nor with the term "biotechnology."

**0052** Processing a query with the term "biotechnology" in a traditional database system 4 would require that the user already has detailed information concerning the knowledge of Mustermann. Furthermore, the term "biotechnology" would have to be enqueued explicitly in a data record allocated to the person Mustermann.

**0053** A query with the term "knowledge" in principle would not make sense in a traditional database system 4 since the abstract term "knowledge" cannot be allocated to a concrete fact "biotechnology."

**0054** In contrast, the computer system 1 according to the invention links abstract terms such as classes and/or attributes with the aid of rules, which provide new characteristic variables as output variables, as for the case at hand. These, in turn, can form abstract variables that can be researched directly by the user. The inference unit 5 then automatically allocates concrete values directly to the abstract variables of the set of rules.

**0055** The example shows that, compared to traditional database systems 4, considerably less pre-knowledge, and

thus also less data input, is required for the computer system 1 according to the invention to arrive at precise search results.

**0056** The object models, as well as the rules allocated to the inference unit 5, can be specified via the editor, at least in part, wherein the user can operate the editor via the terminal 6 that is allocated to the central computer.

**0057** The editor has programming surfaces and/or graphic surfaces, which the user can use to generate rules and/or class structures either in components or completely.

**0058** The configuration of a class structure, for example, can be generated graphically, and the class structure is then created with the aid of graphic elements. Figure 2 shows one example for a graphic representation of class structures of this type.

**0059** Alternatively, the class structure can be created by programming in the respective components.

**0060** For the object models according to Figure 2, for example, it means that corresponding hierarchically divided directories are set up in the editor for the individual classes.

**0061** A directory structure that corresponds to the class structures according to Figure 2 is shown in the following table:

Table 1:

persons

non-employees

employees

clerical employees

technical employees

publications

lectures

documents

**0062** For this, the main directory corresponding to the "persons" class is divided into sub-directories for "non-employees" and "employees." In accordance with the allocation of the subordinate classes "clerical employees" and "technical employees" to the superior class "employees," two corresponding sub-directories are allocated to the "employees" directory in Table 1.

**0063** Attributes can subsequently be allocated to the classes of a class structure generated in this way.

**0064** The programming surface in this case may be designed, for example, such that when clicking on a class of a class

structure, a window is opened up into which the individual attributes are entered as variables. For example, if the user clicks on the "person" class by clicking on the corresponding directory, attributes such as first name, last name, birth date and the like can be input into the corresponding window. In addition, value ranges are allocated to the individual attributes.

**0065** The user can specify complete class structures in this way, to which the individual data are subsequently allocated, wherein the data can be input via the editor. Central input/output units can furthermore be provided for the data input.

**0066** The user can furthermore generate rules with the aid of the editor, which are evaluated in the inference unit 5 for generating the output variables.

**0067** A first option for generating the rules requires the user to freely program the rules, where the rules are in the DAML-L programming language. The user in this case must be proficient in this programming language.

**0068** The rules can be generated more easily if a predetermined number of axioms is defined in the editor, in which the axioms are displayed for the user in a window on

the programming surface of the editor. A defined rule type is allocated in this case to each axiom.

**0069** The user selects a specific axiom by clicking on the programming surface, which generates a rule of the corresponding rule type.

**0070** The axioms preferably are formed with defined mathematical terms.

**0071** Binary relationships such as symmetrical, anti-symmetrical, asymmetrical, inverse, reflexive, non-reflexive or transitive relationships are examples of mathematically defined axioms of this type. Axioms can also express the disjunctive nature of classes.

**0072** Finally, the user can also generate the rules graphically. A suitable graphic surface for this is shown in Figure 3.

**0073** Figure 3 shows a state of the graphic surface where three different windows are opened, wherein each window contains a specific class.

**0074** Two of the windows are identical and relate to the "person" class. A third window relates to the "document" class, wherein the classes are components of the class structure shown in Figure 2.

**0075** According to the example shown in Figure 3, personal data such as name, e-mail address, telephone number and address are allocated as attributes to the "person" class. Furthermore, the relationship "knows of" or the equivalent "has knowledge of" is allocated as attribute to the "person" class.

**0076** The following relationships (connections) are further allocated to the "person" class:

project

is the author of

directs the project

**0077** Characteristic variables of documents, such as title and subject of a document, are allocated to the "document" class as attributes. The following relationships are furthermore allocated to the "document" class:

has date

has author

has a purpose

**0078** Several components of the classes shown in Figure 3 are connected with lines in order to graphically generate a rule, as shown in Figure 3. For the present example, a first line connects the relationship "is author of" in the "person" class to the "document" class. A second line

connects the attribute "on the subject" in the "document" class to the attribute "has knowledge of" in the "person" class.

**0079** The graphic elements for generating the rule are designed such that circular surfaces are provided at the end points of the first line and on one end point of the second line. These circular surfaces point to attributes, classes or relationships which define the requirements of a rule. An arrow is located at the second end point of the second line. The attribute "has knowledge of" in the "person" class, to which the arrow points, forms the conclusion of the rule.

**0080** Thus, the following rule is generated by specifying the two graphic elements according to Figure 3, namely two lines delimited by circular surfaces and/or an arrow.

**0081** "If a person is the author of a document on a subject, then the person has knowledge of this subject."

**0082** A check is finally made via the editor to determine whether the components of class structures and/or rules that are input by the user are correct. In particular, a check is made for consistency and completeness. In case of an incorrect user input, it is advantageous if an error message is generated in the editor.

**0083** The invention has been described in detail with respect to preferred embodiments, and it will now be apparent from the foregoing to those skilled in the art that changes and modifications may be made without departing from the invention in its broader aspects. The invention, therefore, as defined in the appended claims, is intended to cover all such changes and modifications as fall within the true spirit of the invention.